OIPE JC59
JUN 16 2004
PATENT & TRADEMARK OFFICE



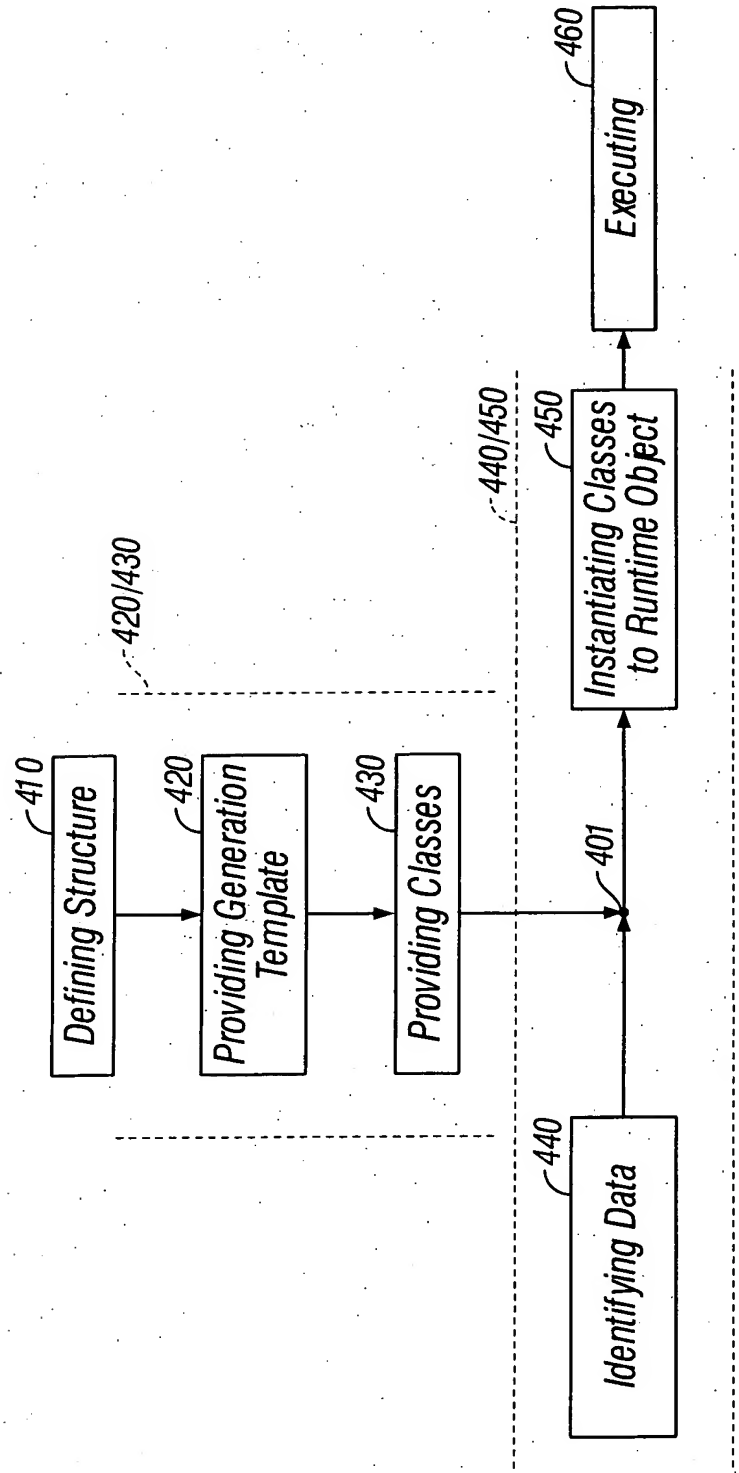


FIG. 1

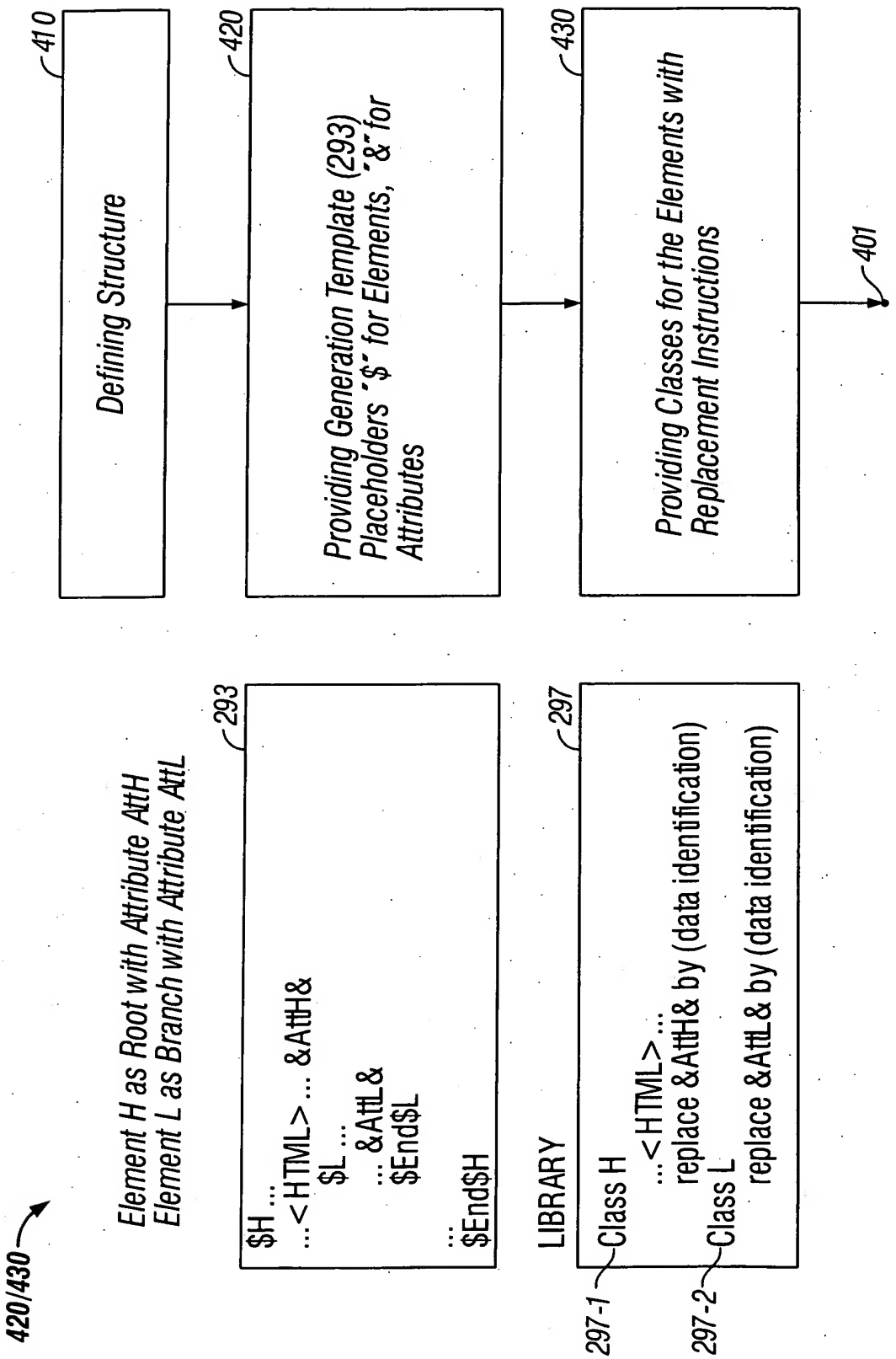
410
Defining Structure
420
Providing Generation Template (293)
Placeholders "$" for Elements, "&" for Attributes
430
Providing Classes for the Elements with Replacement Instructions
401
420/430
Element H as Root with Attribute AttH
Element L as Branch with Attribute AttL
293
$H ...
...<HTML>... &AttH&
$L ...
... &AttL&
$End$L
...
$End$H
297
LIBRARY
297-1
Class H
...<HTML>...
replace &AttH& by (data identification)
297-2
Class L
replace &AttL& by (data identification)

FIG. 2

3/18

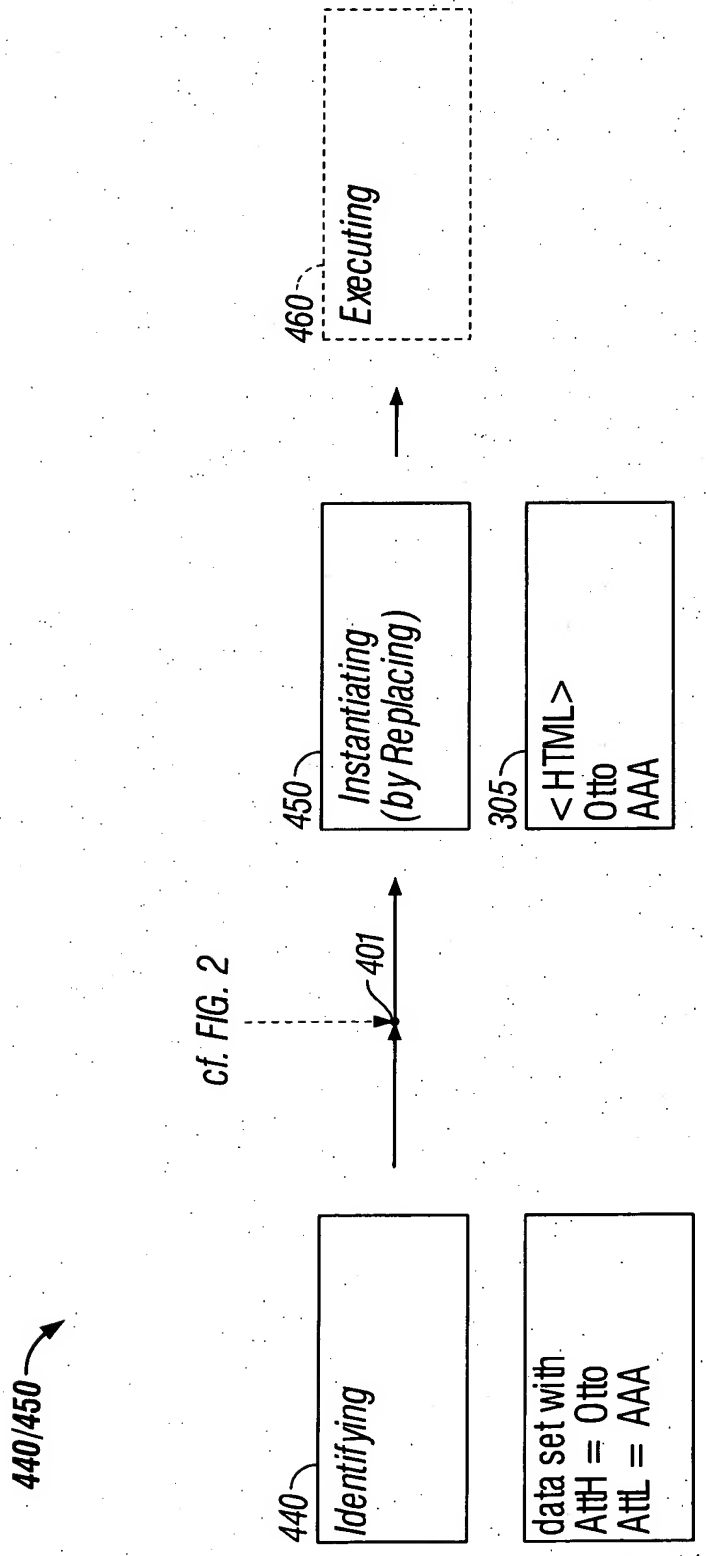
440/450
440
Identifying
data set with
AttH = Otto
AttL = AAA
cf. FIG. 2
401
450
Instantiating
(by Replacing)
305
<HTML>
Otto
AAA
460
Executing

FIG. 3

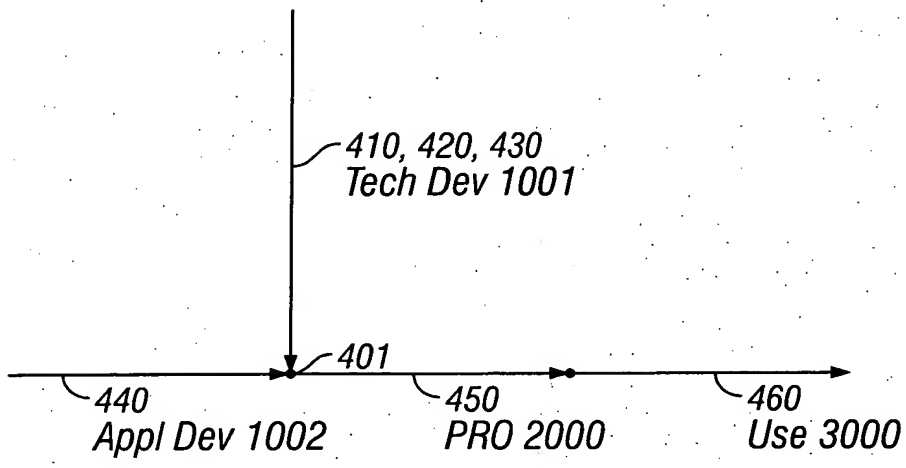
410, 420, 430
Tech Dev 1001
401
440
Appl Dev 1002
450
PRO 2000
460
Use 3000

FIG. 4

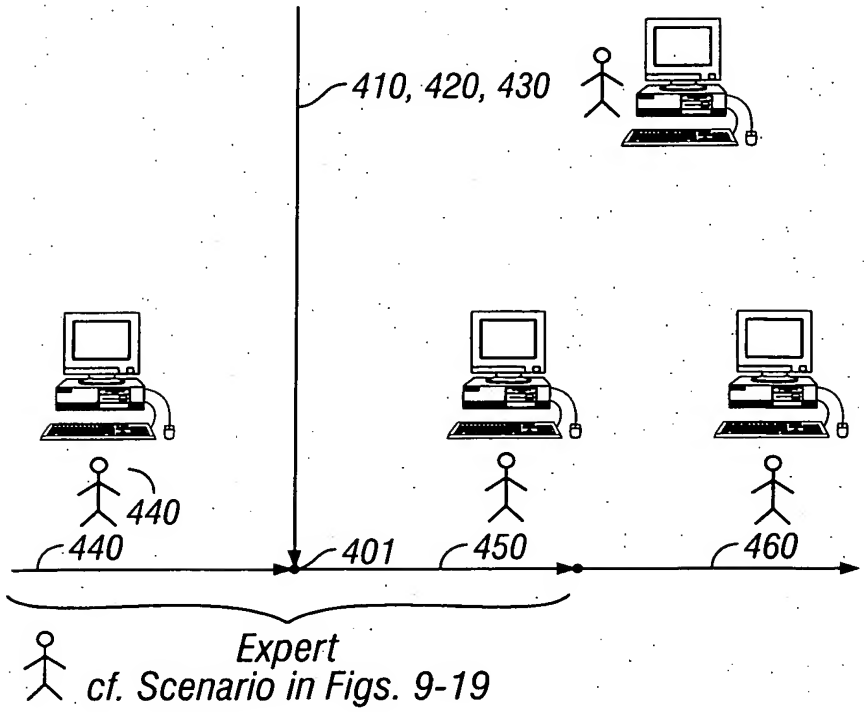
410, 420, 430
440
440
401
450
460
Expert
cf. Scenario in Figs. 9-19

FIG. 5

5/18

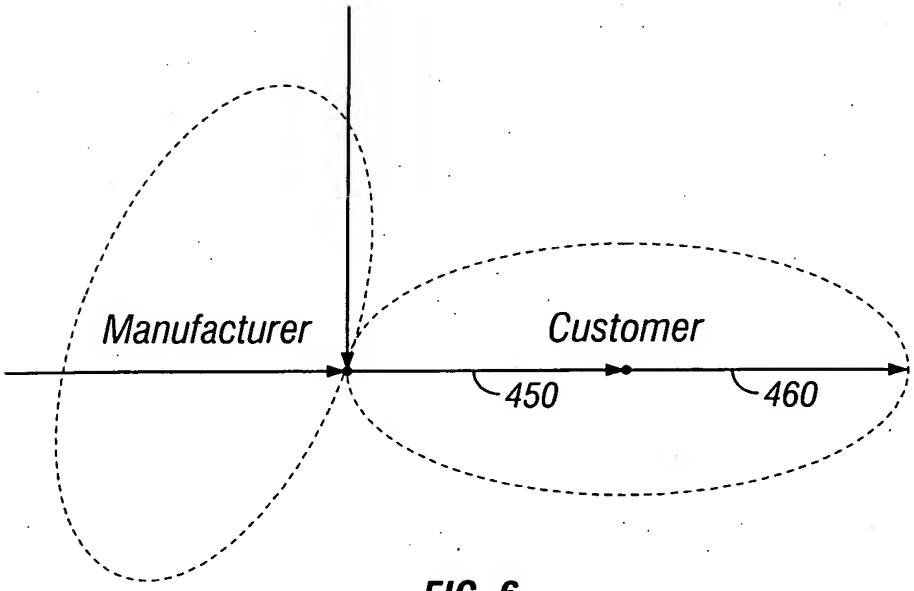


FIG. 6

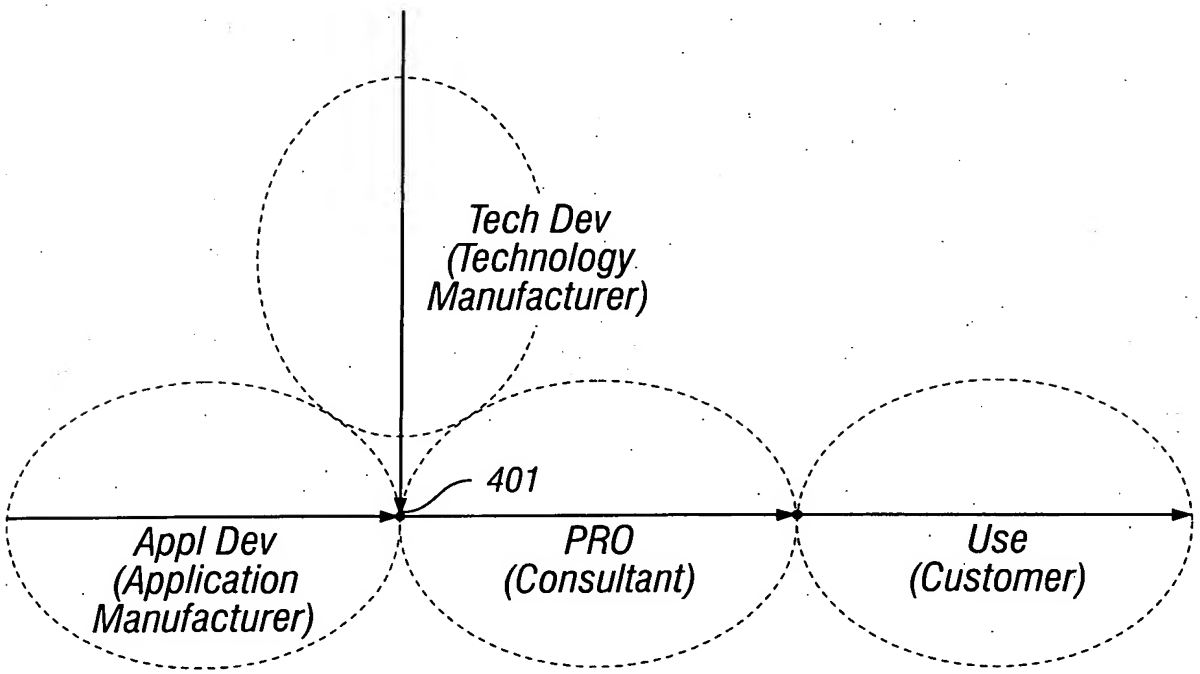


FIG. 7

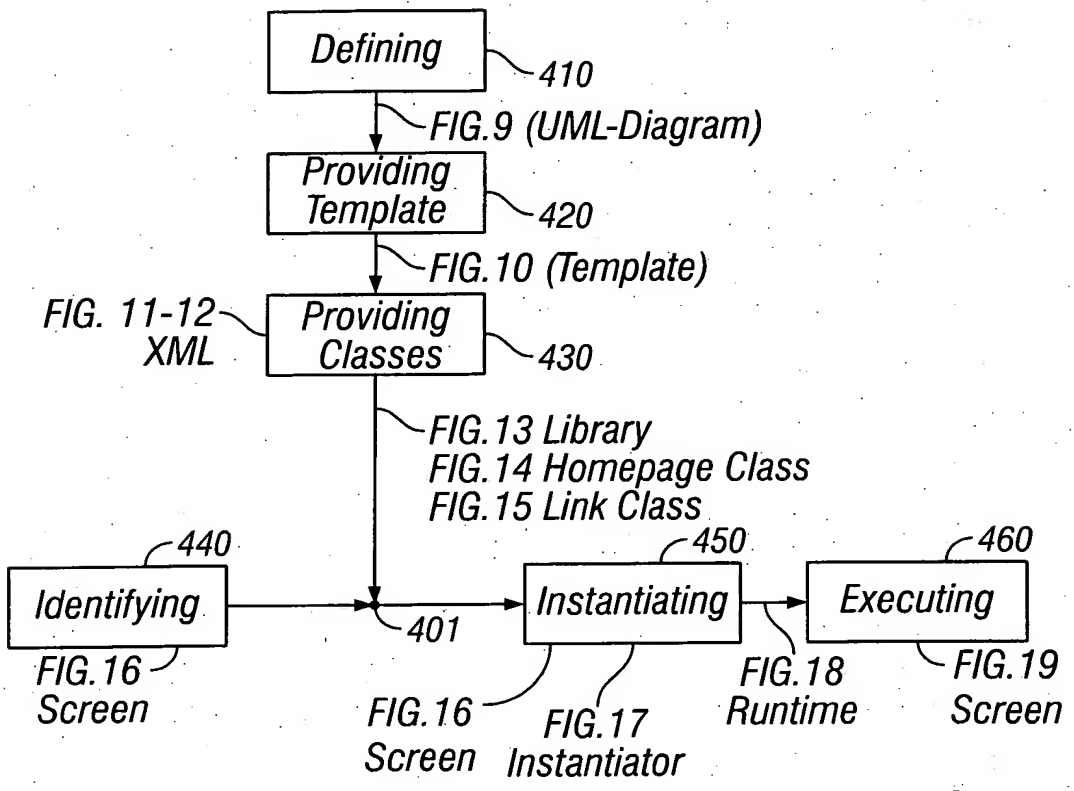
Defining
410
FIG.9 (UML-Diagram)
Providing Template
420
FIG.10 (Template)
FIG. 11-12 XML
Providing Classes
430
FIG.13 Library
FIG.14 Homepage Class
FIG.15 Link Class
440
Identifying
FIG.16 Screen
401
450
Instantiating
FIG.16 Screen
FIG.17 Instantiator
460
Executing
FIG.18 Runtime
FIG.19 Screen

FIG. 8

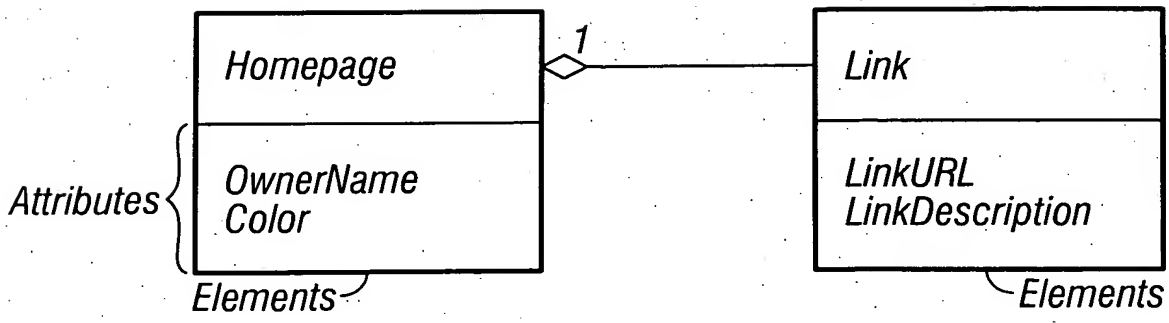
Homepage
1
Link
Attributes
OwnerName
Color
LinkURL
LinkDescription
Elements
Elements

FIG. 9

7/18

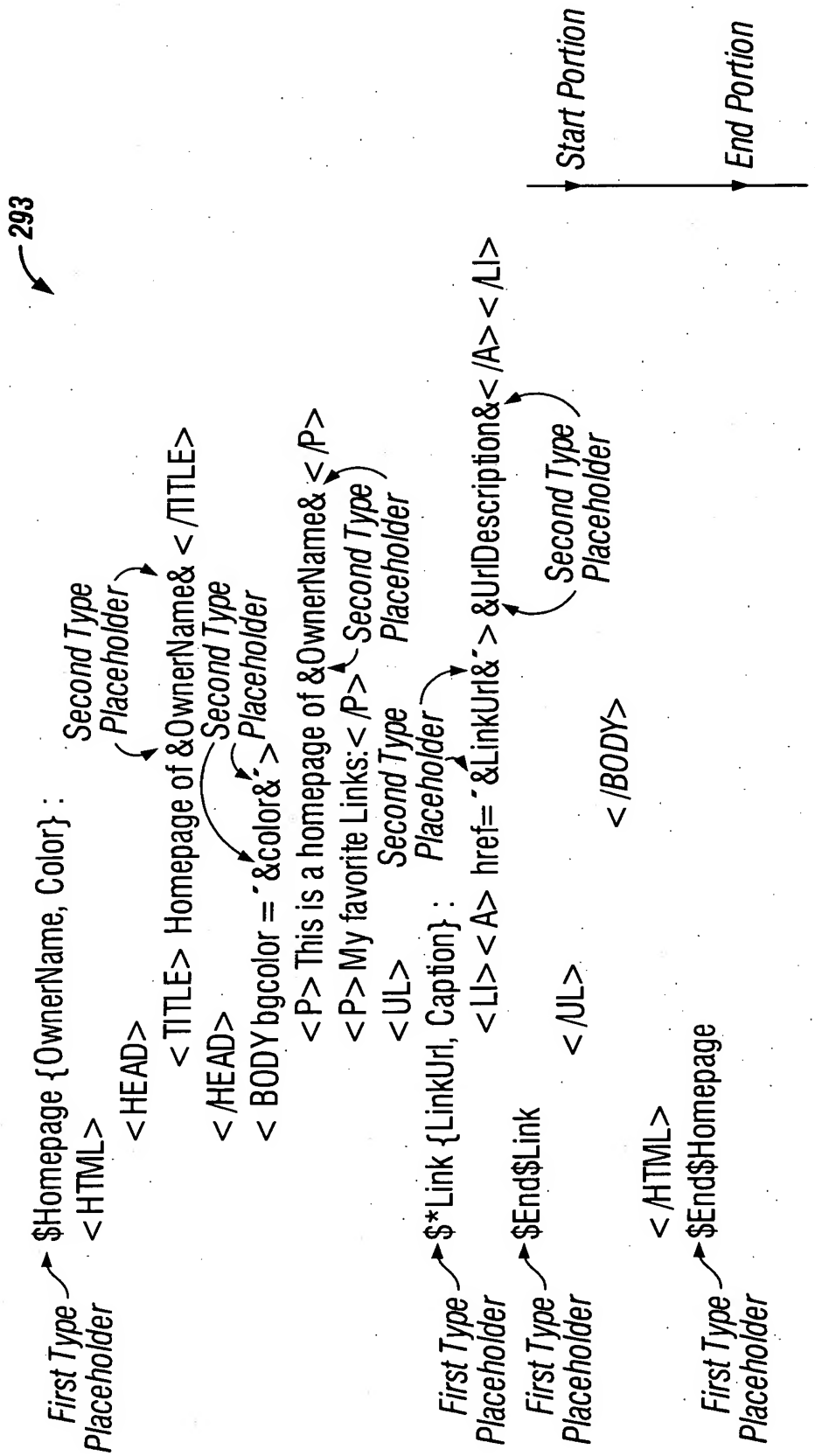
293
First Type Placeholder
$Homepage {OwnerName, Color} :
<HTML>
<HEAD>
<TITLE> Homepage of &OwnerName& </TITLE>
</HEAD>
<BODY bgcolor = ' &color& ' >
<P> This is a homepage of &OwnerName& </P>
<P> My favorite Links: </P>
<UL>
First Type Placeholder
$*Link {LinkUrl, Caption} :
<LI> <A> href= ' &LinkUrl& ' > &UrlDescription& </A> </LI>
First Type Placeholder
$End$Link
</UL>
</BODY>
</HTML>
First Type Placeholder
$End$Homepage
Second Type Placeholder
Second Type Placeholder
Second Type Placeholder
Second Type Placeholder
Start Portion
End Portion

FIG. 10

8/18

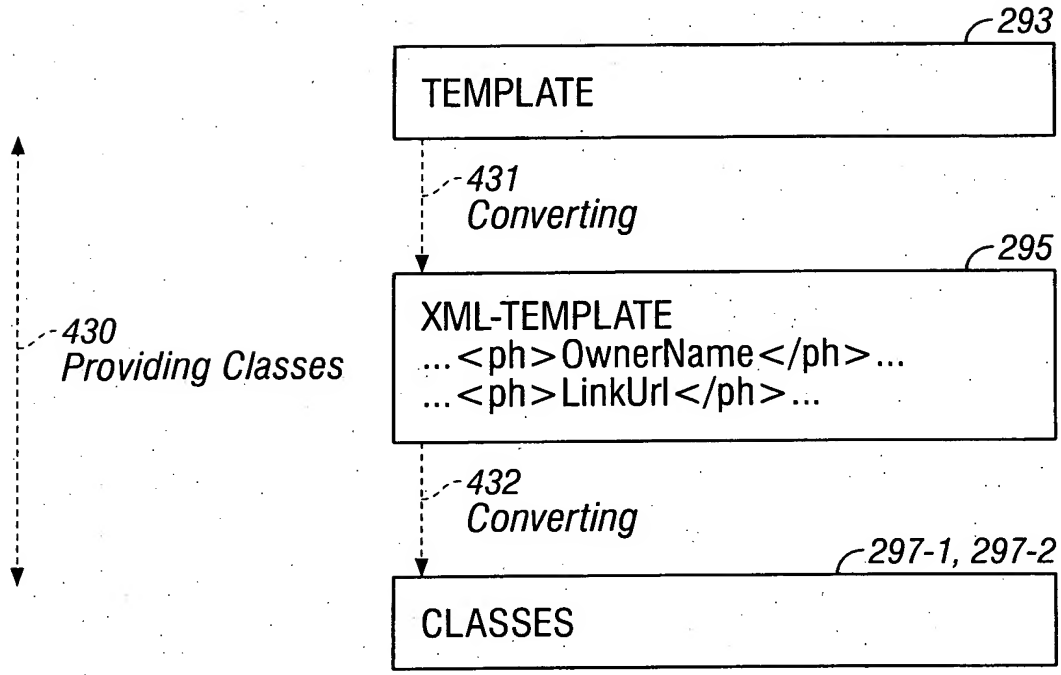
293
TEMPLATE
431
Converting
295
XML-TEMPLATE
...<ph>OwnerName</ph>...
...<ph>LinkUrl</ph>...
430
Providing Classes
432
Converting
297-1, 297-2
CLASSES

FIG.11

295

```
01  <?xml version="1.0"?>                          <library>
    <production name="Homepage">                   <param name="OwnerName"/>
03  <param name="Color"/>                          <template>
    <![CDATA[<HTML>
05      <HEAD><TITLE>Homepage of <ph>OwnerName</ph></TITLE></HEAD>
        <BODY bgcolor="<ph>Color</ph>">
07          <P>This is a homepage of <ph>OwnerName</ph></P>
            <P>My favorite Links:</P>
09          <UL>
    <pph>Link</pph>
11          </UL>
        </BODY>  </HTML>
13  ]]>
    </template>                    <production name="Link" Cardinality="multiple">
15  <param name="LinkUrl"/>    <param name="LinkDescription"/>
    </template>
17  <![CDATA[                      <LI><A href="<ph>LinkUrl</ph>"><ph>LinkDescription
                                   </ph></A></LI>
    ]]>
19  </template>  </production>    </production>      </library>
```

FIG. 12

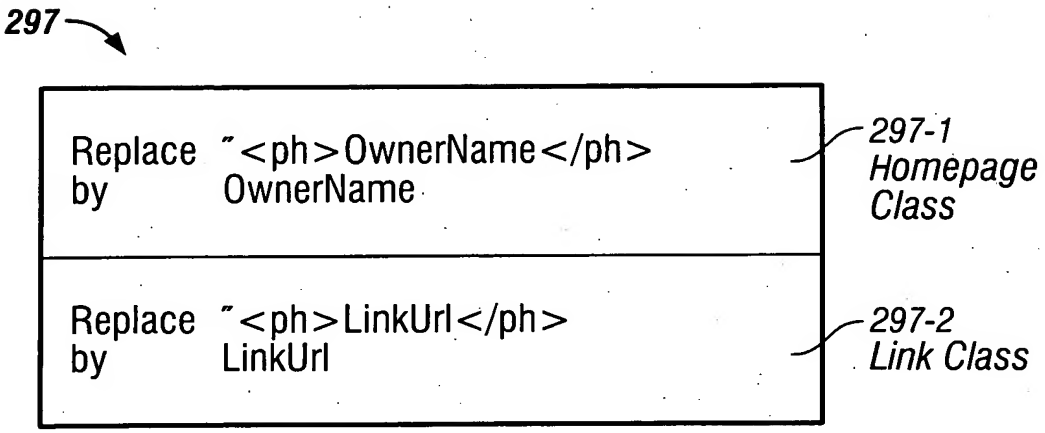
297
Replace by
"<ph>OwnerName</ph>
OwnerName
297-1
Homepage
Class
Replace by
"<ph>LinkUrl</ph>
LinkUrl
297-2
Link Class

FIG. 13

297-1

```
1   Option Explicit                                  Implements IASTNode
    Public OwnerName As String                       Public Color As String
3   Private mLinks As Collection
    Private Property Get Template() As String
5           Template = ""<HTML> Ã°    <HEAD> Ã°              <TITLE> Homepage of
    <ph> OwnerName< /ph> < /TITLE> Ã°    < /HEAD> Ã°      <BODY bgcolor= Ã± < ph> Color
                                                         < /ph> Ã± > Ã°
7   <P> This is a homepage of < ph> OwnerName< /ph> < /P> Ã°          <P> My favorite
    Links:< /P> Ã°              <UL> Ã°< pph> Link< /pph> Ã°          < /UL> Ã°
9   < /BODY> Ã°< /HTML> Ã°"
    End Property
11
Replace
INST
    Public Function Render() As String
13      Dim theCode As String, innerCode As String
        Dim colItem As IASTNode
15
        theCode = Template
17      innerCode = ""
        For Each colItem In mLinks
19              innerCode = innerCode & colItem.Render()
        Next
21      theCode = Replace(theCode, "< pph> Link< /pph>", innerCode)
        theCode = Replace(theCode, "< ph> OwnerName< /ph>", OwnerName)
23      theCode = Replace(theCode, "< ph> Color< /ph>", Color)
        theCode = Replace(theCode, "Ã°",  vbNewLine)
25      theCode = Replace(theCode, "Ã±", """" )
        Render = theCode                                          End Function
```

FIG. 14A

297-1 (Cont.)

```
27  Private Function IASTNode_Render( ) As String
        IASTNode_Render = Render( )
29  End Function

31  Public Function AddLink (ByVal pLinkUrl As String, ByVal pUrlDescription As String) As
    ASTLink
33      Dim 1Link As ASTLink
        Set 1Link = New ASTLink
35      1Link.Initialize pLinkUrl, pUrlDescription
        mLinks.Add 1Link
37      Set AddLink = 1Link
        Set 1Link = Nothing
39  End Function

41  Public Sub Initialize (ByVal pOwnerName As String, ByVal pColor As String)
        OwnerName = pOwnerName
43      Color = pColor                  End Sub

45  Private Sub Class_Initialize( )
        Set mLinks = New Collection     End Sub
47
    Private Sub Class_Terminate( )
49      Set mLinks = Nothing                    End Sub
```

FIG. 14B

297-2

```
1   Option Explicit                 Implements IASTNode

3   Public LinkUrl          As String
    Public UrlDescription   As String
5   Private Property Get Template() As String
        Template = "                    <LI><A
7   href=Ã±<ph>LinkUrl</ph>Ã±><ph>UrlDescription</ph></A></LI>Ã°"   End Property

9   Public Function Render() As String
        Dim theCode As String, innerCode As String
11      Dim colItem As IASTNode
Replace-
ment 13 theCode = Template
        theCode = Replace(theCode, "<ph>LinkUrl</ph>", LinkUrl)
15      theCode = Replace(theCode, "<ph>UrlDescription</ph>", UrlDescription)
        theCode = Replace(theCode, "Ã°", vbNewLine)
17      theCode = Replace(theCode, "Ã±", """")
        Render = theCode                                        End Function
19
    Private Function IASTNode_Render() As String
21      IASTNode_Render = Render()                              End Function
    Public Sub Initialize (ByVal pLinkUrl As String, ByVal pUrlDescription As String)
23      LinkUrl = pLinkUrl
        UrlDescription = pUrlDescription                        End Sub
```

FIG. 15

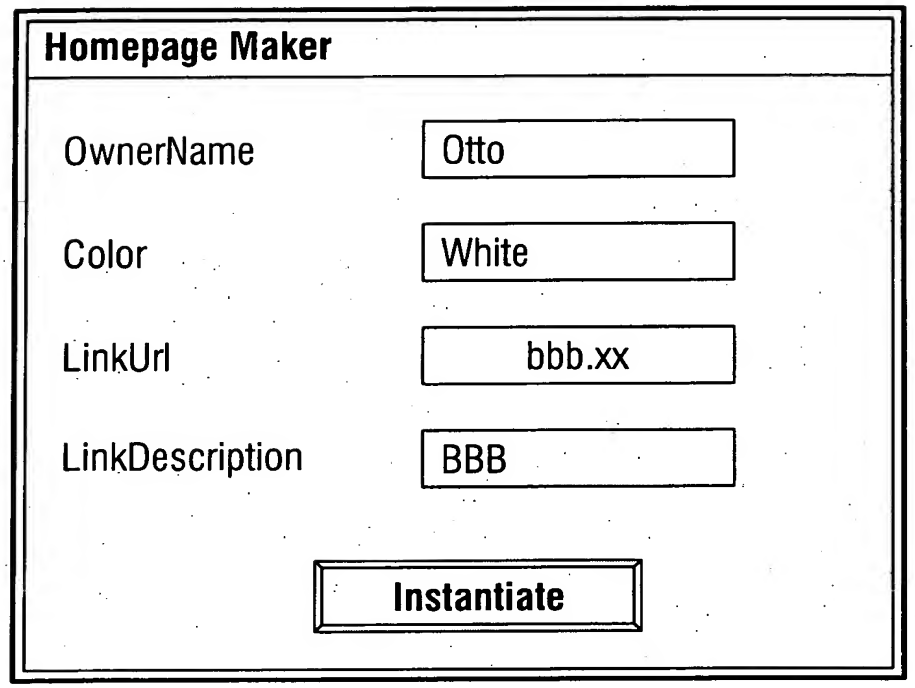
Homepage Maker
OwnerName
Otto
Color
White
LinkUrl
bbb.xx
LinkDescription
BBB
Instantiate

FIG. 16

250

```
Private Sub rtnInstantiate_Click()          ← Activation by Expert

    Dim objAST As ASTHomepage
    Dim i         As Long
    Dim sBuf    As String
    Dim objTS   As TextStream

    Set objAST = New ASTHomepage
    Call objAST.Initialize(txtOwnerName.Text, txtColor.Text)

    For i = 0 To lstLinks.ListCount - 2
        objAST.AddLink lstLinks.List(i+1), lstLinks.List(i)
    Next                                                        } Calling "Replace"
    sBuf = objAST.Render

    Set objTS = mPSO.CreateTextFile(txtOwnerName.Text & ".html")   ← 305 Identification of Runtime Objects
    objTS.Write sBuf
    objTS.Close

    MsgBox "Done!"

End Sub
```

FIG. 17

**16/18**

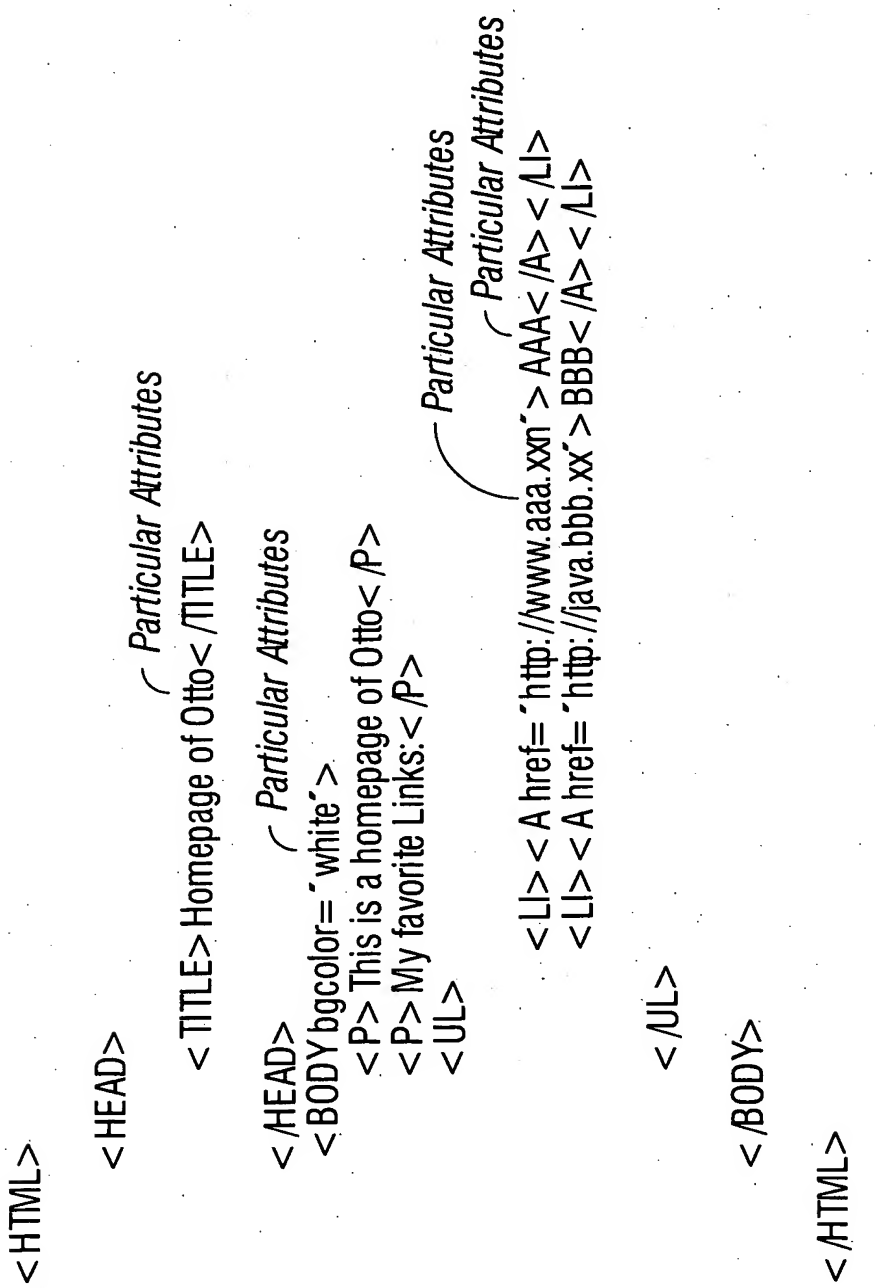
<HTML>
<HEAD>
<TITLE>Homepage of Otto</TITLE>
Particular Attributes
</HEAD>
<BODY bgcolor="white">
Particular Attributes
<P>This is a homepage of Otto</P>
<P>My favorite Links:</P>
<UL>
Particular Attributes
Particular Attributes
<LI><A href="http://www.aaa.xxx">AAA</A></LI>
<LI><A href="http://java.bbb.xx">BBB</A></LI>
</UL>
</BODY>
</HTML>

FIG. 18

17/18

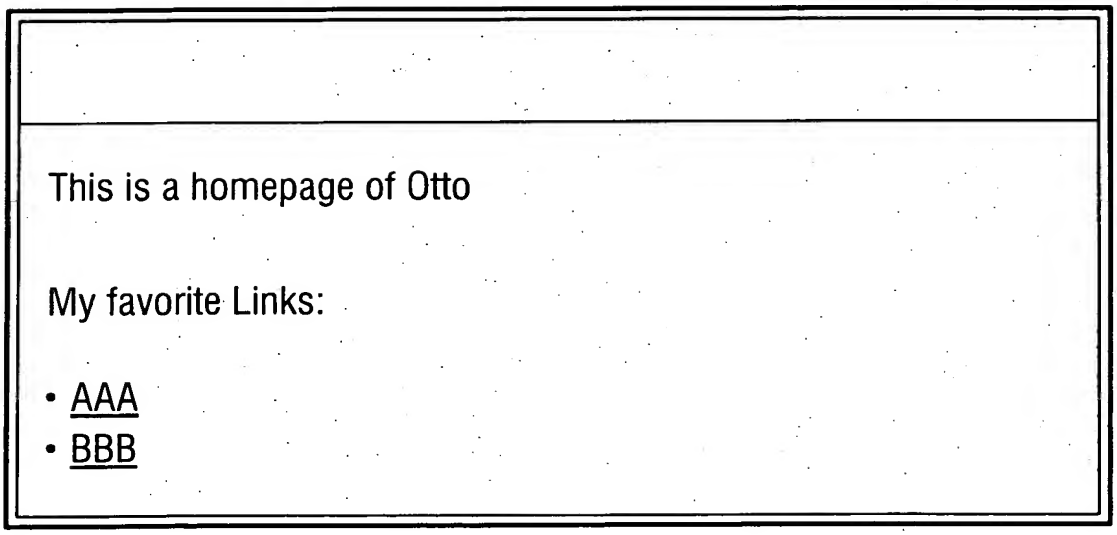


FIG. 19

**18/18**

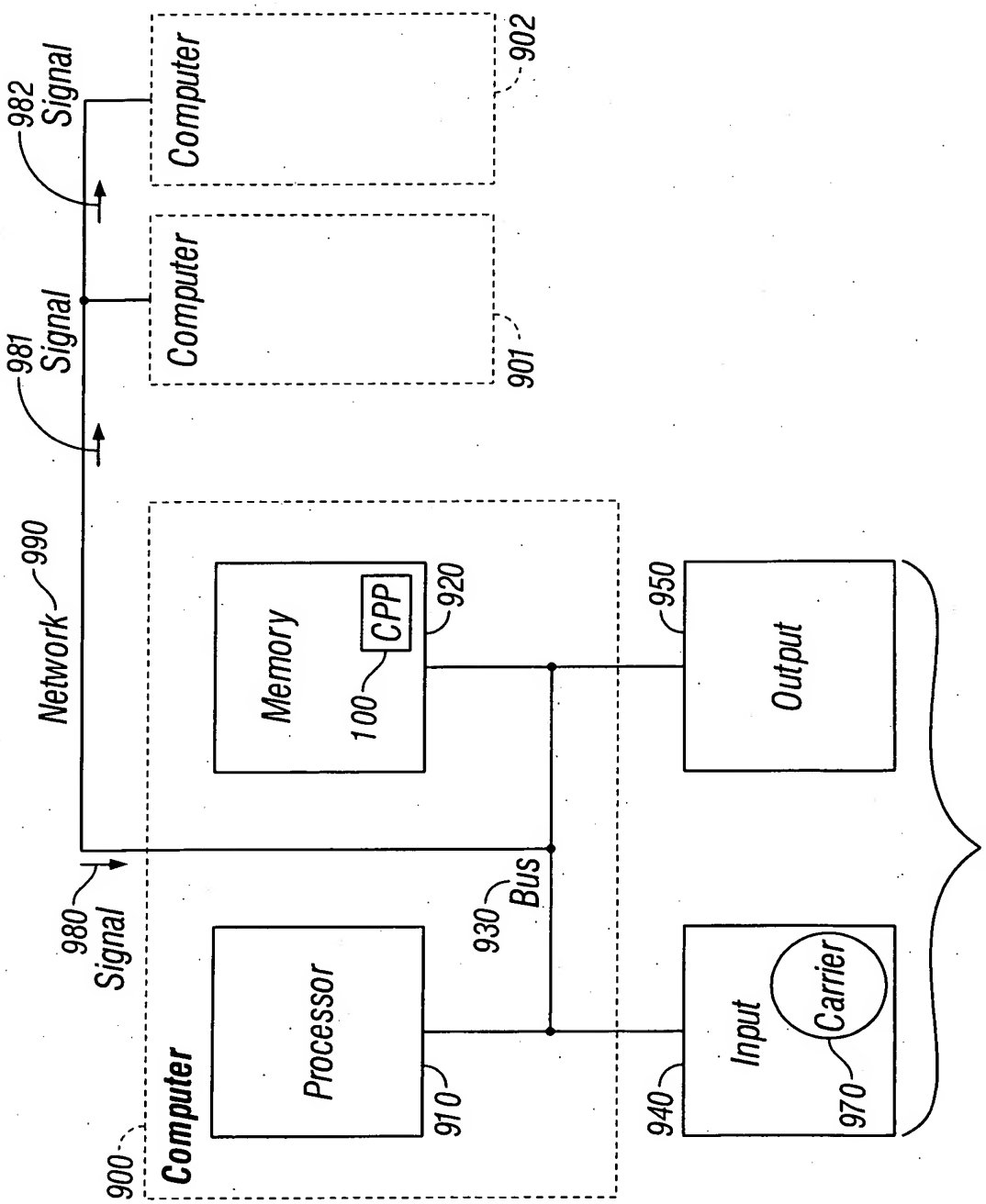
Computer
902
982 Signal
Computer
901
981 Signal
990 Network
Memory
CPP
100
920
950
Output
980 Signal
930 Bus
User Interface
960
Computer
900
Processor
910
940
Input
Carrier
970

FIG. 20